Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

Bereich	Berufsnummer				IHK-Nummer			Prüflingsnummer				
6	7	1	2	0	1							
Sp. 1-2		Sp. 3-6				Sp. 7-9			Sp. 10-14			

Termin: Mittwoch, 29. November 2023

IHK

Abschlussprüfung Winter 2023/24

1201

2 Entwicklung und Umsetzung von Algorithmen

Fachinformatiker
Fachinformatikerin
Anwendungsentwicklung

Teil 2 der Abschlussprüfung

4 Aufgaben
mit Belegsatz
90 Minuten Prüfungszeit
100 Punkte

Bearbeitungshinweise

1. Bevor Sie mit der Bearbeitung der Aufgaben beginnen, überprüfen Sie bitte die **Vollständigkeit** dieses Aufgabensatzes. Die Anzahl der zu bearbeitenden Aufgaben ist auf dem Deckblatt links angegeben. Wenden Sie sich bei Unstimmigkeiten sofort an die Aufsicht, weil Reklamationen am Ende der Prüfung nicht anerkannt werden können.
2. Füllen Sie zuerst die **Kopfzeile** aus. Tragen Sie Ihren Familiennamen, Ihren Vornamen und Ihre Prüflings-Nr. in die oben stehenden Felder ein.
3. Lesen Sie bitte den **Text** der Aufgaben ganz durch, bevor Sie mit der Bearbeitung beginnen.
4. Halten Sie sich bei der Bearbeitung der Aufgaben genau an die **Vorgaben der Aufgabenstellung** zum Umfang der Lösung. Wenn z. B. vier Angaben gefordert werden und Sie sechs Angaben anführen, werden nur die ersten vier Angaben bewertet.
5. Tragen Sie die frei zu formulierenden **Antworten dieser offenen Aufgaben** in die dafür lt. Aufgabenstellung vorgesehenen Bereiche (Lösungszeilen, Formulare, Tabellen u. a.) des Arbeitsbogens ein.
6. Sofern nicht ausdrücklich ein Brief oder eine Formulierung in ganzen Sätzen gefordert werden, ist eine **stichwortartige Beantwortung** zulässig.
7. Schreiben Sie deutlich und gut lesbar. Ein nicht eindeutig zuzuordnendes oder **unleserliches Ergebnis** wird als **falsch** gewertet.
8. Zur Lösung der Rechenaufgaben darf ein nicht programmierter, netzunabhängiger **Taschenrechner** ohne Kommunikationsmöglichkeit mit Dritten verwendet werden.
9. Wenn Sie ein **gerundetes Ergebnis** eintragen und damit weiterrechnen müssen, rechnen Sie (auch im Taschenrechner) nur mit diesem gerundeten Ergebnis weiter.
10. Für **Hilfsaufzeichnungen** können Sie das in der Tasche beigelegte Konzeptpapier verwenden. Bewertet werden jedoch grundsätzlich nur Ihre Eintragungen in diesem Aufgabensatz.

Wird vom Korrektor ausgefüllt!

Bewertung

Für die Bewertung gilt die Vorgabe der Punkte in den Lösungshinweisen.

1. Aufg. ☐☐ Punkte (15 16) 2. Aufg. ☐☐ Punkte (17 18) 3. Aufg. ☐☐ Punkte (19 20) 4. Aufg. ☐☐ Punkte (21 22)

Prüfungszeit ☐ (23)
Die entsprechende Ziffer (1, 2 oder 3) finden Sie in der Abfrage nach der Prüfungszeit im Anschluss an die letzte Aufgabe.

Gesamtpunktzahl ☐☐☐ (24 25 26)

Prüfungsort, Datum

Unterschrift

Gemeinsame Prüfungsaufgaben der Industrie- und Handelskammern. Dieser Aufgabensatz wurde von einem überregionalen Ausschuss, der entsprechend § 40 Berufsbildungsgesetz zusammengesetzt ist, beschlossen. Hinweis: Im Interesse einer besseren Lesbarkeit wird in der Aufgabenstellung und in den Angaben zur Aufgabenstellung nur die männliche Form (generisches Maskulinum) verwendet. Die verkürzte Sprachform beinhaltet keine Wertung und die gewählten männlichen Formulierungen gelten uneingeschränkt auch für die weiteren Geschlechter.
Die Vervielfältigung, Verbreitung und öffentliche Wiedergabe der Prüfungsaufgaben und Lösungen ist nicht gestattet. Zuwiderhandlungen werden zivil- und strafrechtlich (§§ 97 ff., 106 ff. UrhG) verfolgt. – © ZPA Nord-West 2023 – Alle Rechte vorbehalten!

Die Aufgaben 1 bis 4 beziehen sich auf die folgende Ausgangssituation:

Die AMAG Soft GmbH hat sich auf Software-Entwicklung im gesamten Umfeld vom Anbau bis zum Vertrieb von Weinprodukten spezialisiert.

Über lokale Messstellen im Weinberg und autonome Drohnen können Daten, wie zum Beispiel Niederschläge, Temperatur und Blattfärbungen ermittelt werden.

Korrekturrand

1. Aufgabe (25 Punkte)

Um auf die Güte des Weins schließen zu können, soll eine App entwickelt werden, die anhand von Messungen an den Trauben vor der Weinlese über die Güte Auskunft geben kann.

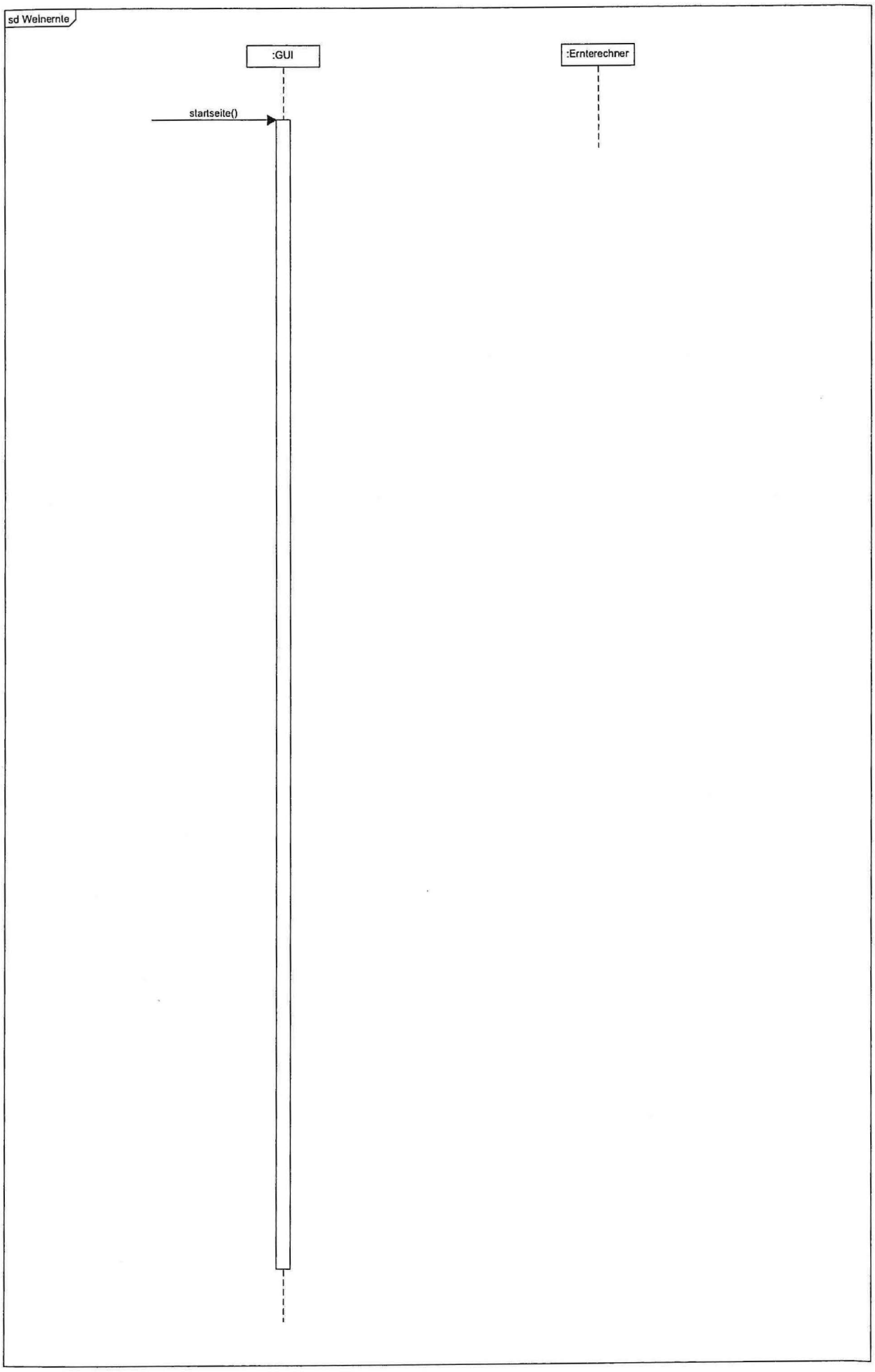
Nach dem Aufruf der App kann der zuvor gemessene Säuregehalt der Trauben eingegeben werden (Methode einlesenSaeuregehalt() der Klasse GUI).

Ist der Säuregehalt zu hoch, werden keine weiteren Messungen mehr vorgenommen.
Es erfolgt die Ausgabe „Säuregehalt zu hoch" (Methode ausgabe("...") der Klasse GUI).
Wenn der Säuregehalt nicht zu hoch ist, wird der Oechslegrad eingegeben (einlesenOechslegrad() der Klasse GUI).

Anschließend erfolgt die Berechnung des Reifegrads durch die Methode berechneReifegrad(oechslegrad) der Klasse Ernterechner.
Falls der Reifegrad größer oder gleich 100 ist, erfolgt die Ausgabe „hoher Reifegrad".
Falls der Reifegrad größer oder gleich 80 (und kleiner als 100) ist, erfolgt die Ausgabe „mittlerer Reifegrad".
Falls der Reifegrad größer oder gleich 70 (und kleiner als 80) ist, erfolgt die Ausgabe „niedriger Reifegrad".

Erstellen Sie auf der Folgeseite für den beschriebenen Sachverhalt ein Sequenzdiagramm.

Korrekturrand



2. Aufgabe (25 Punkte)

Korrekturrand

Für den Flug der Drohnen soll die Reihenfolge bestimmt werden, in der vorgegebene Geopositionen angeflogen werden.

Ausgehend von der aktuellen Position soll immer die Position als nächstes angeflogen werden, die der aktuellen Position am nächsten liegt.

Die anzufliegenden Positionen liegen in einem eindimensionalen Array *geoPositions* als Objekte der Klasse *GeoPos* vor.

GeoPos
- latitude: double - longitude: double - altitude: double

Zu jedem Attribut sind öffentliche Zugriffsmethoden (set/get) vorhanden.

Für Entfernungsberechnungen zwischen Geopositionen kann die Methode *getDistance* der Klasse *GeoCalculator* verwendet werden:

GeoCalculator
+ getDistance(pos1: GeoPos, pos2: GeoPos): double { static }

Der Algorithmus für den Drohnenflug soll sein Ergebnis in einem Array *flightPositions* speichern und zurückgeben. Die Arbeitsweise des Algorithmus *calculateFlight*() wird wie folgt beschrieben:

- Das Array *geoPositions* kann durch den Algorithmus verändert werden.
- Die Geoposition mit dem Index 0 des Arrays *geoPositions* wird zur ersten aktuellen Position. Diese Position wird im Array *flightPositions* gespeichert (Startposition!) und kann dann aus dem Array der noch anzufliegenden Positionen (*geoPositions*) gelöscht werden. Hinweis: bereits implementiert, siehe Vorgabe.
- Solange noch Geopositionen im Array *geoPositions* vorhanden sind:

  Ermitteln der Position im Array *geoPositions* mit der kürzesten Entfernung zur aktuellen Position.
  Diese Position wird im Array *flightPositions* gespeichert und zur neuen aktuellen Position.
  Diese Position kann dann aus dem Array *geoPositions* gelöscht werden.

Hinweis: Der größte Doublewert kann mit Double.*MAX_VALUE* abgerufen werden.

Vervollständigen Sie dahingehend den Pseudocode auf der gegenüberliegenden Seite.

```
calculateFlight(geoPositions: GeoPos[] ) : GeoPos[]
    flight_positions = new GeoPos[geoPositions.length] // Array der angeflogenen
                                                       // Positionen
    flightPositions[0] = geoPositions[0]               // Position 0 als Startposition
    currentPos = geoPositions[0]
    geoPositions.remove(0)                             // aus dem Array der noch
                                                       // anzufliegenden Positionen
                                                       // entfernen
```

Korrekturrand

3. Aufgabe (25 Punkte)

Korrekturrand

a) Für den Weinberg sollen die Messstellen sowie die Messgeräte und deren Messdaten in einer relationalen Datenbank verwaltet werden. Als Grundlage zur Entwicklung des relationalen Datenmodells liegen nachfolgende Angaben vor:

Die Messstellen sind an verschiedenen Stellen auf dem Weinberg verteilt. Die Position einer Messstelle wird genau durch ihre Geokoordinaten (Längen- und Breitengrad sowie die Höhe) beschrieben. Jede Messstelle liefert Messwerte von verschiedenen Messgrößen wie z. B. Messgröße = Temperatur, Messwert = 22, Einheit = °C. Für die einzelnen Messwerte sollen weiterhin auch noch der Zeitpunkt der Messung (Datum, Uhrzeit) und die Messstelle erfasst werden.

An jeder Messstelle können mehrere Messgeräte eingesetzt werden, welche die Messwerte erfassen und weiterleiten. Dabei ist nur relevant, welches Messgerät aktuell an welcher Messstelle eingesetzt wird. Historische Daten spielen keine Rolle. Für jedes Messgerät sollen folgende Angaben erfasst werden: Bezeichnung, Herstellungsdatum, Hersteller, Anschaffungsdatum und das Datum der letzten Überprüfung.

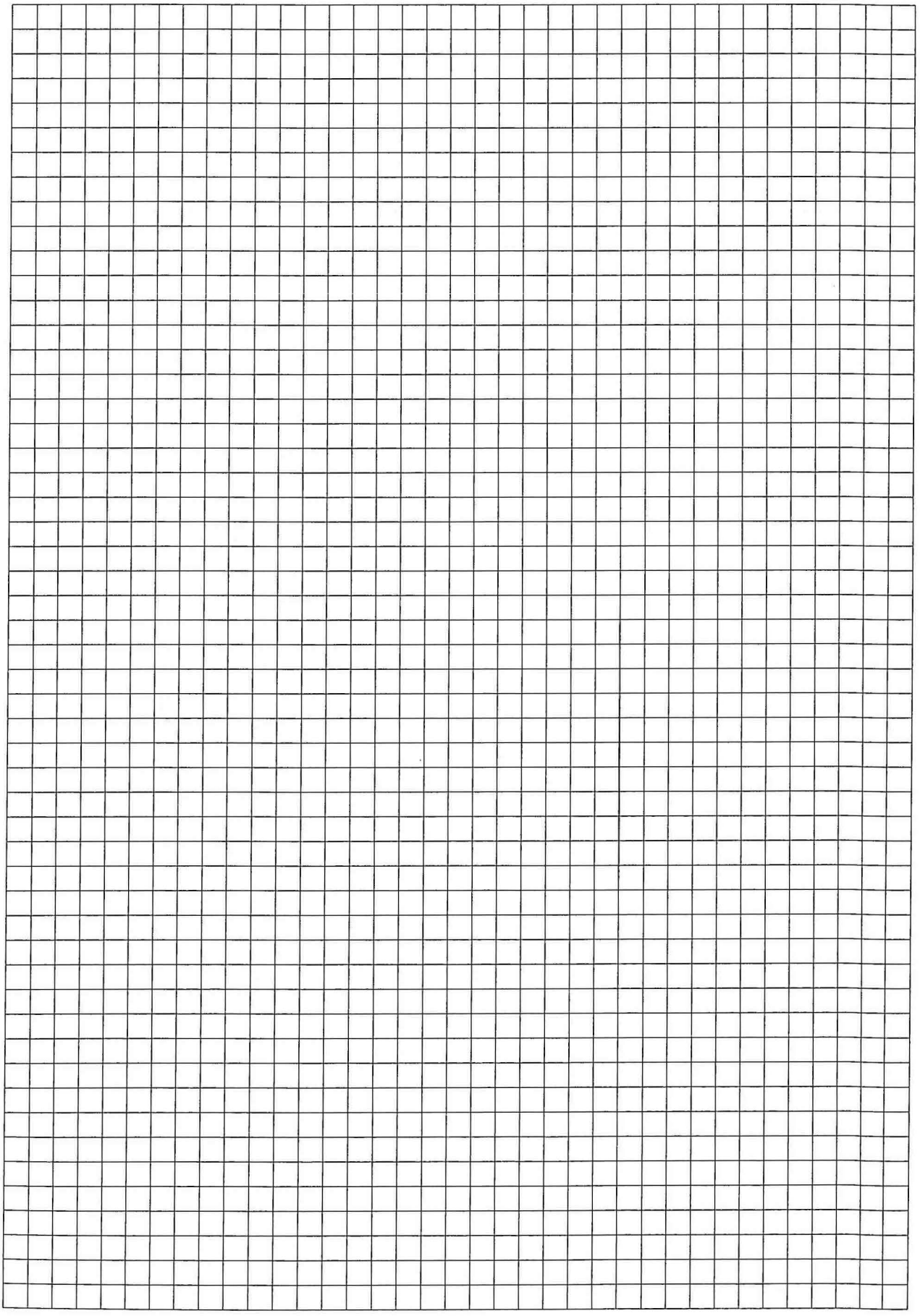
Erstellen Sie aus diesen Angaben für die Messwerterfassung ein relationales Datenmodell in der dritten Normalform. Geben Sie Kardinalitäten an und kennzeichnen Sie die Primärschlüssel mit PK und die Fremdschlüssel mit FK.

18 Punkte

b) Weiterhin sollen Bilder der autonomen Drohnen für die weitere Verarbeitung und Auswertung z. B. durch eine KI unkomprimiert in einem „Raw"-Format gespeichert werden. Die Drohnen fliegen einmal pro Woche über den Weinberg und machen pro Flug jeweils 200 Bilder. Die Kamera liefert Bilder mit 20 Megapixeln. Die Farben pro Pixel werden im RGB-Format abgespeichert, wobei pro Farbkanal 8 Bit verwendet werden.

Korrekturrand

Berechnen Sie den benötigten Speicherbedarf in GiB, welcher für die Bilder innerhalb eines Jahres (52 Wochen) anfällt. Runden Sie auf ganze GiB. Der Rechenweg ist anzugeben. 7 Punkte



4. Aufgabe (25 Punkte)

Korrekturrand

Für die Kundenverwaltung und die Warenwirtschaft sind folgende Tabellen vorhanden.

Tabelle: Kunde

Kd_IdKey	Kd_Firma	Kd_Strasse	Kd_PLZ	Kd_Ort	Kd_Nummer
1	LikeLimo	Musterstr. 12	50778	Köln	012204
2	Gasthaus 'Die Perle'	Perlenstr. 22	50778	Köln	012201
3	Traberstübchen	Traberweg 1	50889	Köln	012205
...					

Tabelle: Artikel

Art_IdKey	Art_Nummer	Art_Bezeichnung	Art_Preis	Art_VkEinheit	Art_MwStSatz	Art_Jahrgang	Art_WAIdKey
8	GB-12555	Grauburgunder	15.20	Flasche	19.00	2005	1
9	GB-12556	Grauburgunder	13.10	Flasche	19.00	2006	1
10	GB-12558	Grauburgunder	11.25	Flasche	19.00	2008	1
11	WH-23	Weißherbst	16.40	Flasche	19.00	2004	1
...							

Tabelle: WeinArt

WA_IdKey	WA_Weinart
1	Weißwein
2	Rotwein
3	Rosewein
4	Schaumwein
...	

Tabelle: Rechnung

Rg_IdKey	Rg_KdIdKey	Rg_Nummer	Rg_Datum	Rg_ZahlFristTage
2223	2	RG-002249	2023-05-02	14
2224	3	RG-002250	2023-05-02	14
2225	3	RG-002251	2023-05-04	14
2226	1	RG-002252	2023-05-05	7
...				

Tabelle: RechnungPosition

RgPos_IdKey	RgPos_RgIdKey	RgPos_ArtIdKey	RgPos_Nr	RgPos_Mg	RgPos_EinzelPreis	RgPos_RabattProzent	RpPos_MwStSatz
555434	2223	14	1	4	13.80	0.00	19.00
555435	2223	19	2	12	17.20	0.00	19.00
555436	2223	10	3	6	11.25	5.00	19.00
555437	2223	11	4	12	16.40	5.00	19.00
555438	2224	14	1	8	13.80	0.00	19.00
555439	2225	10	1	6	11.25	0.00	19.00
555450	2225	11	2	6	16.40	0.00	19.00
555451	2225	12	3	12	15.40	8.00	19.00
555452	2225	13	4	6	16.40	0.00	19.00
555453	2226	19	1	6	17.20	0.00	19.00
...							

a) Sie sollen eine Kundenliste erstellen, in der alle Kunden ausgegeben werden, welche aus dem PLZ-Gebiet 50800 – 51999 ansässig sind.

Erstellen Sie dazu eine SQL-Anweisung, welche eine Ergebnistabelle entsprechend dem folgenden Beispiel ausgibt. 4 Punkte

Kd_IdKey	Kd_Firma	Kd_Strasse	Kd_PLZ	Kd_Ort	Kd_Nummer
3	Traberstübchen	Traberweg 1	50889	Köln	012205
6	Rheinkneipe	Am Rhein 2	50888	Köln	014333
...					

b) Sie möchten eine Auflistung mit dem niedrigsten und dem höchsten Artikelpreis sowie die Anzahl der Weine pro Jahrgang sortiert nach Jahrgang absteigend.

Korrekturrand

Erstellen Sie dazu eine SQL-Anweisung, welche eine Ergebnistabelle entsprechend dem folgenden Beispiel ausgibt. 5 Punkte

Jahrgang	NiedrigsterPreis	HöchsterPreis	Anzahl der Weine je Jahrgang
2008	11.25	13.80	2
2007	14.00	14.00	1
...			

c) Fälschlicherweise wurden für Rotweine bei allen Kunden anstelle von 12 % Rabatt 0 % Rabatt berechnet.

Erstellen Sie eine SQL-Anweisung, mit der Sie für alle Rotweine, die im Zeitraum Mai 2023 berechnet wurden, den Rabatt von 0 % auf 12 % ändern. 8 Punkte

Fortsetzung 4. Aufgabe →

Fortsetzung 4. Aufgabe

Korrekturrand

d) Erstellen Sie eine SQL-Anweisung mit der Sie den berechneten durchschnittlichen Umsatz je Kunde unabhängig der gewährten Rabatte erhalten. 8 Punkte

Beispiel-Ergebnistabelle:

Kd_IdKey	Kd_Firma	Kd_Strasse	Kd_PLZ	Kd_Ort	Kd_Nummer	Umsatz
1	LikeLimo	Musterstr. 12	50778	Köln	012204	149.520000
2	Gasthaus 'Die Perle	Perlenstr. 22	50778	Köln	012201	128.171250
999	Gasthaus zum Fass	Fassstraße 3	99900	Fasshausen	012203	0
...						

PRÜFUNGSZEIT – NICHT BESTANDTEIL DER PRÜFUNG!

Wie beurteilen Sie nach der Bearbeitung der Aufgaben die zur Verfügung stehende Prüfungszeit?

[1] Sie hätte kürzer sein können.
[2] Sie war angemessen.
[3] Sie hätte länger sein müssen.

Abschlussprüfung Winter 2023/24

IHK

Belegsatz

Fachinformatiker/Fachinformatikerin
Anwendungsentwicklung
1201

Teil 2 der Abschlussprüfung

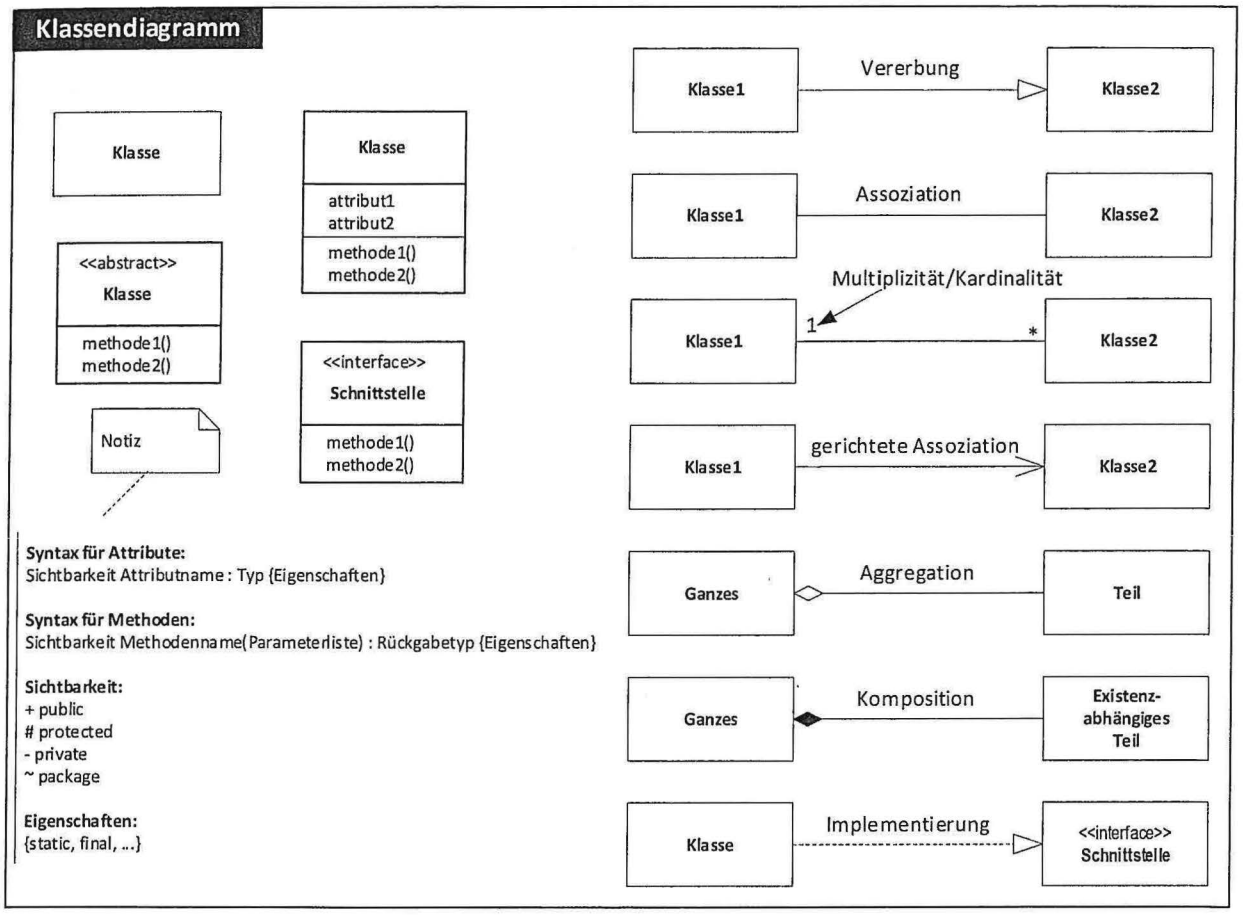
Der Belegsatz ist Grundlage für beide Prüfungsbereiche.

1. Planen eines Softwareproduktes
2. Entwicklung und Umsetzung von Algorithmen

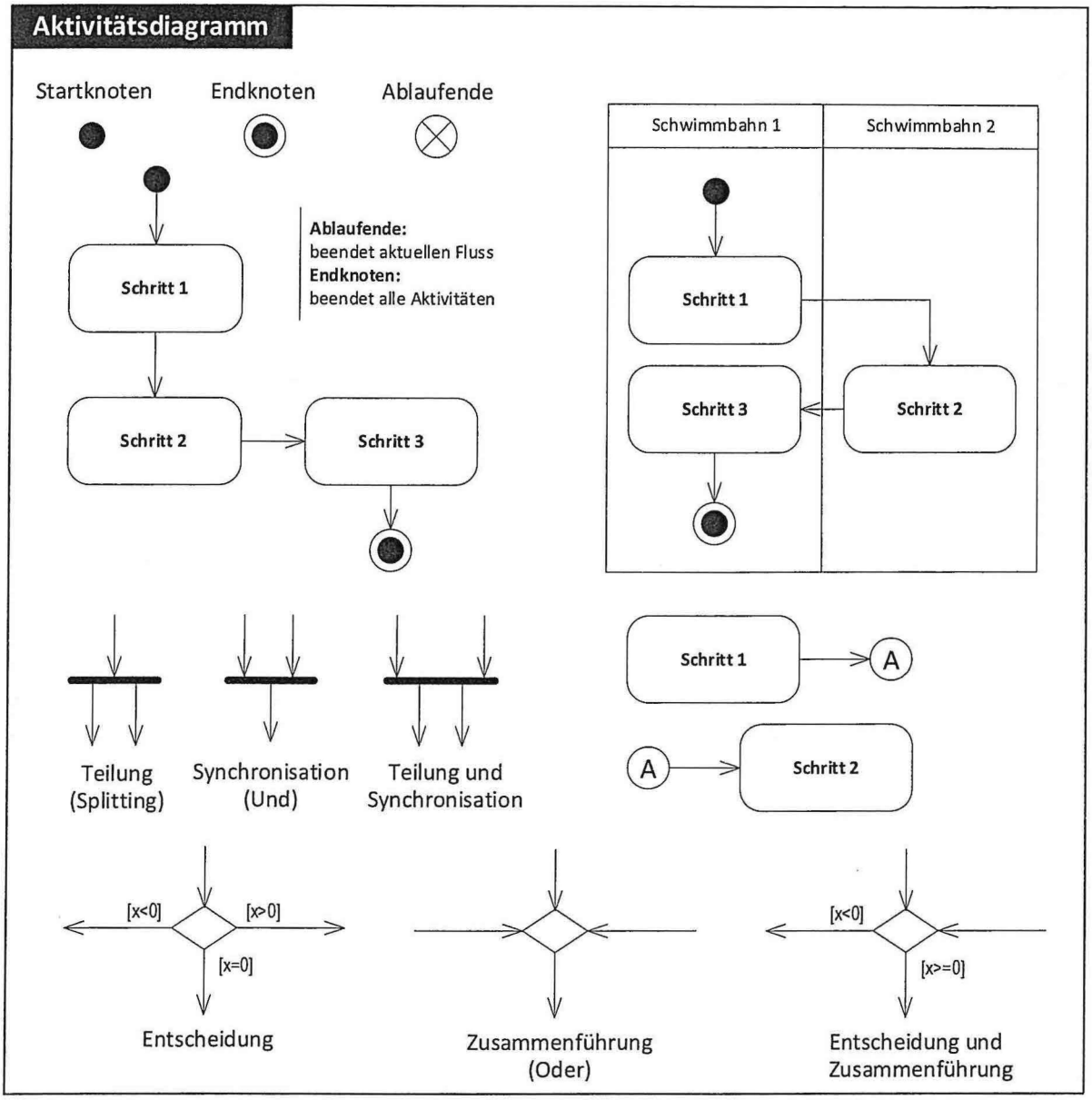
	Seite
UML-Klassendiagramm	2
UML-Aktivitätsdiagramm	3
UML-Sequenzdiagramm	4
UML-Anwendungsfalldiagramm	5
SQL-Syntax (Auszug)	6/7

Gemeinsame Prüfungsaufgaben der Industrie- und Handelskammern. Dieser Aufgabensatz wurde von einem überregionalen Ausschuss, der entsprechend § 40 Berufsbildungsgesetz zusammengesetzt ist, beschlossen. Hinweis: Im Interesse einer besseren Lesbarkeit wird in der Aufgabenstellung und in den Angaben zur Aufgabenstellung nur die männliche Form (generisches Maskulinum) verwendet. Die verkürzte Sprachform beinhaltet keine Wertung und die gewählten männlichen Formulierungen gelten uneingeschränkt auch für die weiteren Geschlechter.
Die Vervielfältigung, Verbreitung und öffentliche Wiedergabe der Prüfungsaufgaben und Lösungen ist nicht gestattet. Zuwiderhandlungen werden zivil- und strafrechtlich (§§ 97 ff., 106 ff. UrhG) verfolgt. – © ZPA Nord-West 2023 – Alle Rechte vorbehalten!

UML-Klassendiagramm

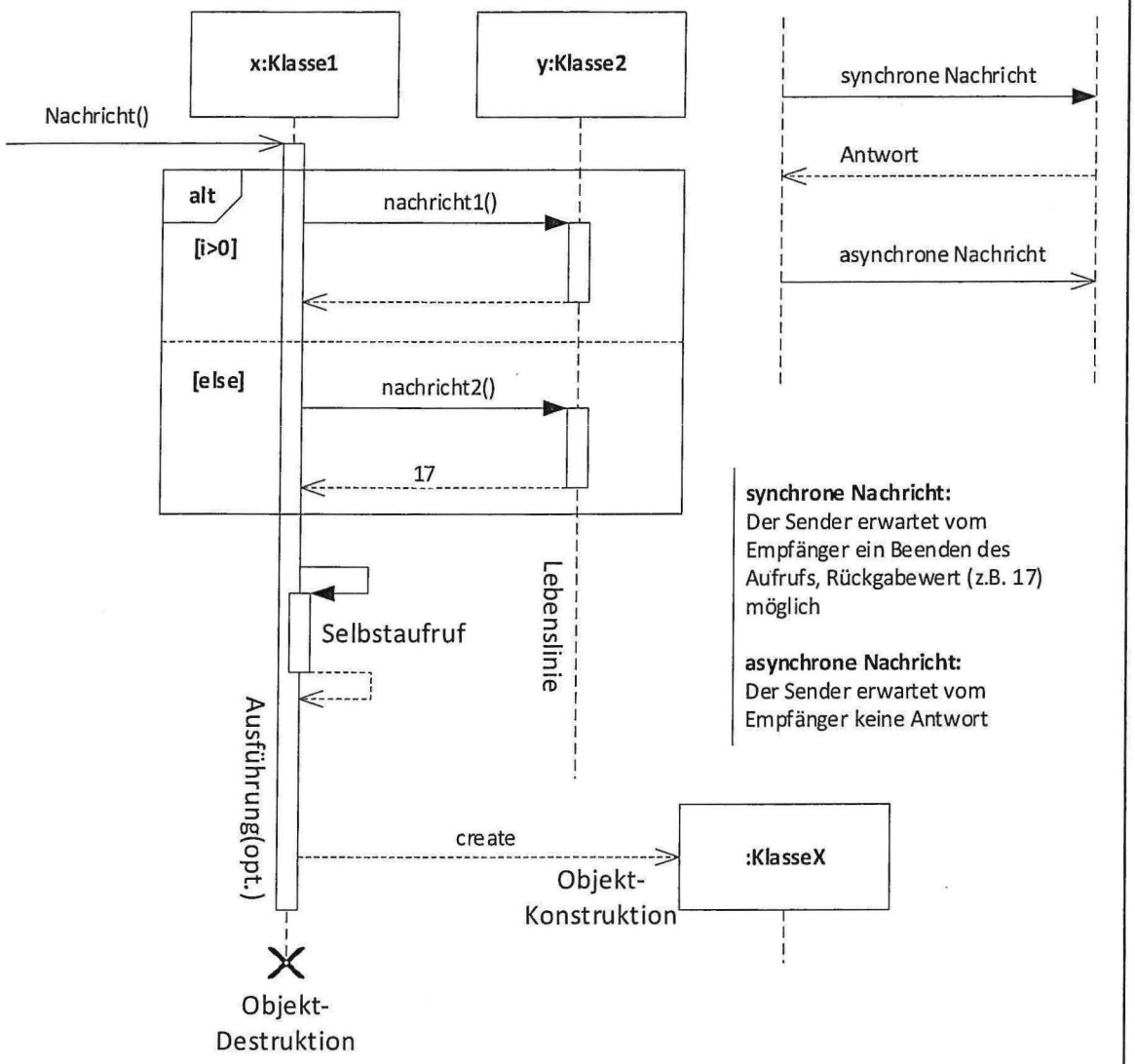


UML-Aktivitätsdiagramm

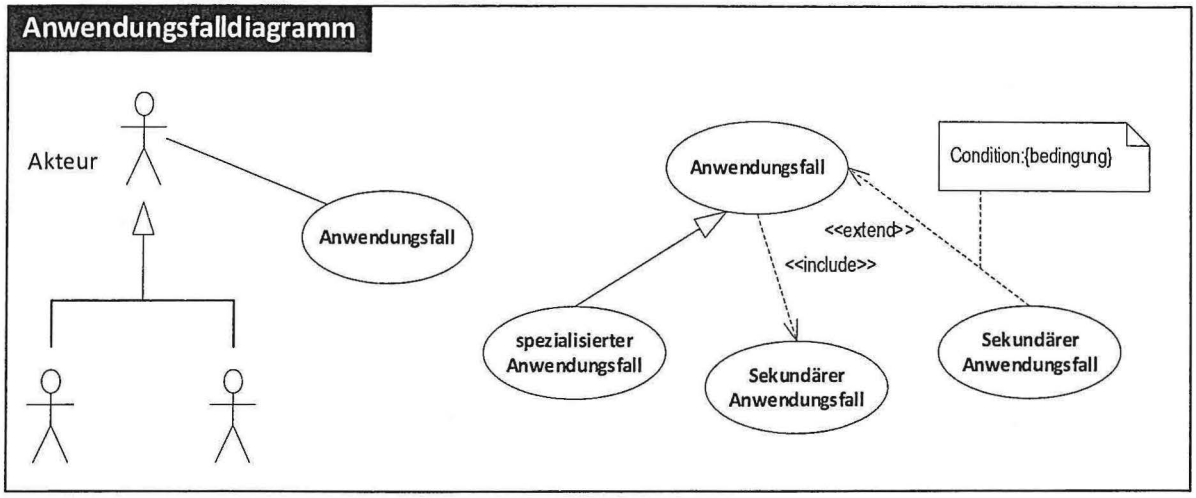


UML-Sequenzdiagramm

Sequenzdiagramm



UML-Anwendungsfalldiagramm



SQL-Syntax (Auszug)

Syntax	Beschreibung
Tabelle	
CREATE TABLE Tabellenname(Spaltenname < DATENTYP >, Primärschlüssel, Fremdschlüssel)	Erzeugt eine neue leere Tabelle mit der beschriebenen Struktur
ALTER TABLE Tabellenname **ADD COLUMN** Spaltenname Datentyp **DROP COLUMN** Spaltenname Datentyp **ADD FOREIGN KEY**(Spaltenname) **REFERENCES** Tabellenname(Primärschlüsselspaltenname)	Änderungen an einer Tabelle: Hinzufügen einer Spalte Entfernen einer Spalte Definiert eine Spalte als Fremdschlüssel
CHARACTER	Textdatentyp
DECIMAL	Numerischer Datentyp (Festkommazahl)
DOUBLE	Numerischer Datentyp (Doppelte Präzision)
INTEGER	Numerischer Datentyp (Ganzzahl)
DATE	Datum (Format DD.MM.YYYY)
PRIMARY KEY (Spaltenname)	Erstellung eines Primärschlüssels
FOREIGN KEY (Spaltenname) **REFERENCES** Tabellenname(Primärschlüsselspaltenname)	Erstellung einer Fremdschlüssel-Beziehung
DROP TABLE Tabellenname	Löscht eine Tabelle
Befehle, Klauseln, Attribute	
SELECT * | Spaltenname1 [, Spaltenname2, ...]	Wählt die Spalten einer oder mehrerer Tabellen, deren Inhalte in die Liste aufgenommen werden sollen; alle Spalten (*) oder die namentlich aufgeführten
FROM	Name der Tabelle oder Namen der Tabellen, aus denen die Daten der Ausgabe stammen sollen
SELECT ... **FROM ...** **(SELECT ...** **FROM ...** **WHERE ...) AS** tbl **WHERE ...**	Unterabfrage (subquery), die in eine äußere Abfrage eingebettet ist. Das Ergebnis der Unterabfrage wird wie eine Tabelle – hier mit Namen "tbl" – behandelt.
SELECT **DISTINCT**	Eliminiert Redundanzen, die in einer Tabellen auftreten können, Werte werden jeweils nur einmal angezeigt.
JOIN / INNER JOIN	Liefert nur die Datensätze zweier Tabellen, die gleiche Datenwerte enthalten
LEFT JOIN / LEFT OUTER JOIN	Liefert von der erstgenannten (linken) Tabelle alle Datensätze und von der zweiten Tabelle jene, deren Datenwerte mit denen der ersten Tabelle übereinstimmen
RIGHT JOIN / RIGHT OUTER JOIN	Liefert von der zweiten (rechten) Tabelle alle Datensätze und von der ersten Tabelle jene, deren Datenwerte mit denen der zweiten Tabelle übereinstimmen
WHERE	Bedingung, nach der Datensätze ausgewählt werden sollen
WHERE EXISTS (subquery) **WHERE NOT EXISTS** (subquery)	Die Bedingungen EXISTS prüft, ob die Suchbedingung einer Unterabfrage mindestens eine Zeile zurückliefert. NOT EXIST negiert die Bedingung.
WHERE ... IN (subquery) **WHERE NOT... IN** (subquery)	Der Wert des Datenfelds ist in der auswählten Menge vorhanden. Der Wert des Datenfelds ist in der auswählten Menge nicht vorhanden.
GROUP BY Spaltenname1 [,Spaltenname2, ...]	Gruppierung (Aggregation) nach Inhalt des genannten Feldes
ORDER BY Spaltenname1 [,Spaltenname2, ...] **ASC | DESC**	Sortierung nach Inhalt des genannten Feldes oder der genannten Felder ASC: aufsteigend; DESC: absteigend

Syntax	Beschreibung
Datenmanipulation	
DELETE FROM Tabellenname	Löschen von Datensätzen in der genannten Tabelle
UPDATE Tabellenname **SET**	Aktualisiert Daten in Feldern einer Tabelle
INSERT INTO Tabellenname[(spalte1, spalte2, ...)] **VALUES** (Wert für Spalte 1 [, Wert für Spalte 2, ...]) oder **SELECT ... FROM ... WHERE**	Fügt Datensätze in die genannte Tabelle, die entweder mit festen Werten belegt oder Ergebnis eines SELECT-Befehls sind
Berechtigungen kontrollieren	
CREATE Benutzer | Rolle **IDENTIFIED BY** 'Passwort'	Erzeugt einen neuen Benutzer oder eine neue Rolle mit einem Passwort
GRANT Recht | Rolle **ON** *.* | Datenbank.* | Datenbank.Objekt **TO** Benutzer | Rolle [WITH GRANT OPTION]	Weist einem Benutzer oder einer Rolle ein Recht auf ein bestimmtes Datenbank-Objekt zu Weist einem Benutzer eine Rolle zu
REVOKE Rechte | Rollen **ON** *.* | Datenbank.* | Datenbank.Objekt **FROM** Benutzer | Rolle	Entzieht einem Benutzer oder einer Rolle ein Recht auf ein bestimmtes Datenbank-Objekt Entzieht einem Benutzer eine Rolle
Aggregatfunktionen	
AVG(Spaltenname)	Ermittelt das arithmetische Mittel aller Werte im angegebenen Feld
COUNT(Spaltenname | *)	Ermittelt die Anzahl der Datensätze mit Nicht-NULL-Werten im angegebenen Feld oder alle Datensätze der Tabelle (dann mit Operator *)
SUM(Spaltenname | Formel)	Ermittelt die Summe aller Werte im angegebenen Feld oder der Formelergebnisse
MIN(Spaltenname | Formel)	Ermittelt den kleinsten aller Werte im angegebenen Feld
MAX (Spaltenname | Formel)	Ermittelt den größten aller Werte im angegebenen Feld
Funktionen	
LEFT(Zeichenkette, Anzahlzeichen)	Liefert *Anzahlzeichen* der Zeichenkette von links.
RIGHT(Zeichenkette, Anzahlzeichen)	Liefert *Anzahlzeichen* der Zeichenkette von rechts.
CURRENT | GETDATE	Liefert das aktuelle Datum mit der aktuellen Uhrzeit
CONVERT(time,[DatumZeit])	Liefert die Uhrzeit aus einer DatumZeit-Angabe
DATE(Wert)	Wandelt einen Wert in ein Datum um
DAY(Datum)	Liefert den Tag des Monats aus dem angegebenen Datum
MONTH(Datum)	Liefert den Monat aus dem angegebenen Datum
TODAY	Liefert das aktuelle Datum
WEEKDAY(Datum)	Liefert den Tag der Woche aus dem angegebenen Datum
YEAR(Datum)	Liefert das Jahr aus dem angegebenen Datum
DATEADD(Datumsteil, Intervall, Datum)	Fügt einem Datum ein Intervall (ausgedrückt in den unter Datumsteil angegebenen Einheiten) hinzu
DATEDIFF(Datumsteil, Anfangsdatum, Enddatum) Datumsteile: **DAY, MONTH, YEAR**	Liefert Enddatum-Startdatum (ausgedrückt in den unter Datumsteil angegebenen Einheiten)
Operatoren	
AND	Logisches UND
LIKE	Überprüfung von Text auf Gleichheit wenn Platzhalter ("regular expressions") eingesetzt werden.
NOT	Logische Negation
OR	Logisches ODER
IS NULL	Überprüfung auf NULL
=	Test auf Gleichheit
>, >=, <, <=, < >	Test auf Ungleichheit
*	Multiplikation
/	Division
+	Addition, positives Vorzeichen
-	Subtraktion, negatives Vorzeichen

Stand 2021-09-30

Abschlussprüfung Winter 2023/24
Lösungshinweise

Fachinformatiker/Fachinformatikerin
Anwendungsentwicklung
1201

2 Entwicklung und Umsetzung von Algorithmen

Teil 2 der Abschlussprüfung

Allgemeine Korrekturhinweise

Die Lösungs- und Bewertungshinweise zu den einzelnen Handlungsschritten sind als Korrekturhilfen zu verstehen und erheben nicht in jedem Fall Anspruch auf Vollständigkeit und Ausschließlichkeit. Neben hier beispielhaft angeführten Lösungsmöglichkeiten sind auch andere sach- und fachgerechte Lösungsalternativen bzw. Darstellungsformen mit der vorgesehenen Punktzahl zu bewerten. Der Bewertungsspielraum des Korrektors (z. B. hinsichtlich der Berücksichtigung regionaler oder branchenspezifischer Gegebenheiten) bleibt unberührt.

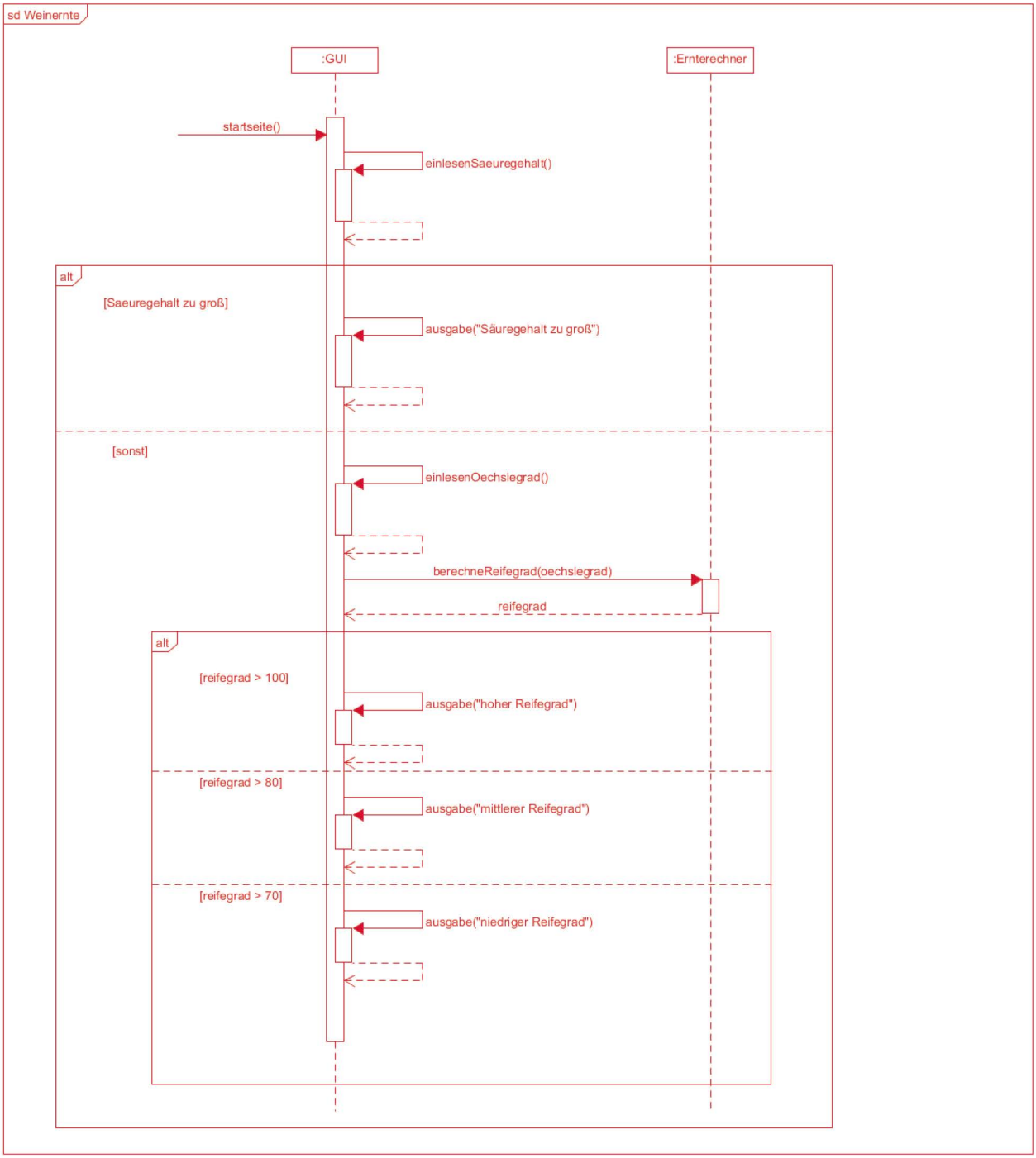
Zu beachten ist die unterschiedliche Dimension der Aufgabenstellung (nennen – erklären – beschreiben – erläutern usw.).

Für die Bewertung gilt folgender Punkte-Noten-Schlüssel:

Note 1	=		100 – 92 Punkte	Note 2	=	unter	92 – 81 Punkte
Note 3	=	unter	81 – 67 Punkte	Note 4	=	unter	67 – 50 Punkte
Note 5	=	unter	50 – 30 Punkte	Note 6	=	unter	30 – 0 Punkte

Die Vervielfältigung, Verbreitung und öffentliche Wiedergabe der Prüfungsaufgaben und Lösungen ist nicht gestattet. Zuwiderhandlungen werden zivil- und strafrechtlich (§§ 97 ff., 106 ff. UrhG) verfolgt. – © ZPA Nord-West 2023 – Alle Rechte vorbehalten!

1. Aufgabe (25 Punkte)

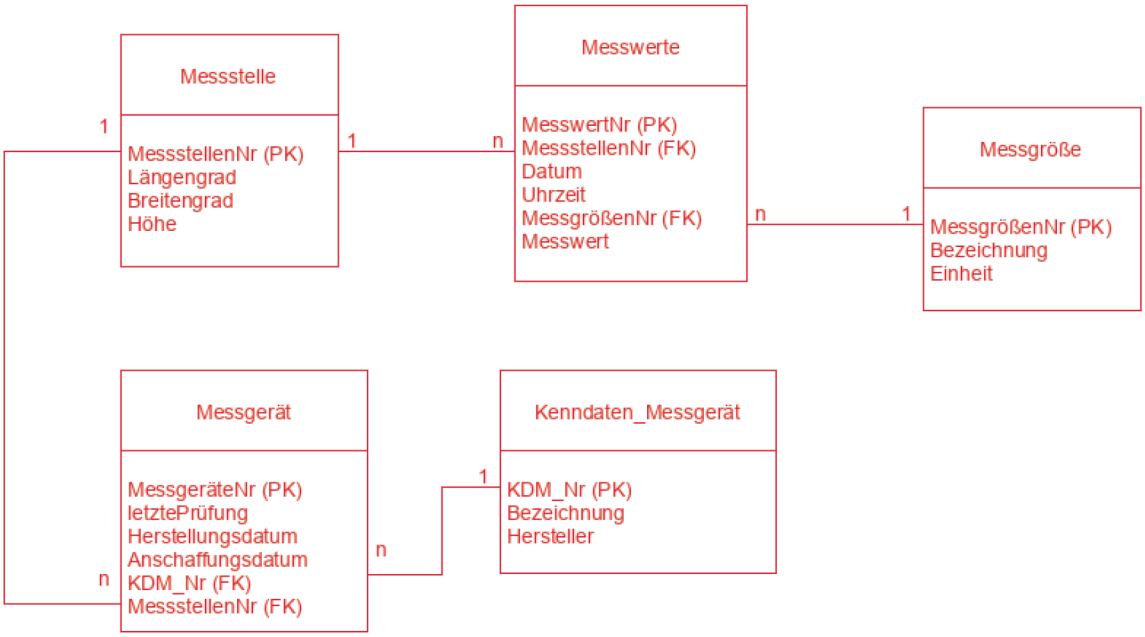


2. Aufgabe (25 Punkte)

```
calculateFlight(geoPositions: GeoPos[] ) : GeoPos[]

flight_positions = new GeoPos[geoPositions.length]  // Array der angeflogenen
                                                    // Positionen
flightPositions[0] = geoPositions[0]                // Position 0 als Startposition
currentPos = geoPositions[0]
geoPositions.remove(0)                              // aus dem Array der noch
                                                    // anzufliegenden Positionen
                                                    // entfernen

next = 1                                        // nächster Index in flightPositions
while (geoPositions.length > 0)                 // Solange noch Positionen anzufliegen sind
       minDistance = Double.MAX_VALUE
       minPos = 0
       for i = 0 to geoPositions.length -1
              distance = geoCalculator.getDistance(currentPos, geoPositions[i])
              if distance < minDistance
                     minDistance = distance
                     minPos = i
              end if
       end for

       flightPositions[next] = geoPositions[minPos]
       next = next + 1
       currentPos = geoPositions[minPos]
       geoPositions.remove(minPos)
end while

return flightPositions

end calculateFlight
```

3. Aufgabe (25 Punkte)

a) 18 Punkte



Je Tabelle mit allen Attributen 1 Punkt (5 Punkte)
Je Primärschlüssel 1 Punkt (5 Punkte)
Je Beziehung mit Kardinalität 1 Punkt (4 Punkte)
Je Fremdschlüssel 1 Punkt (4 Punkte)

b) 7 Punkte

Hinweis: Rundungsdifferenzen können im Ergebnis berücksichtigt werden.
20.000.000 Pixel/Bild * 8 Bit/Farbkanal * 3 Farbkanal/Pixel = 480.000.000 Bit/Bild

480.000.000 Bit/Bild / 8 Bit/Byte = 60.000.000 Byte/Bild / 1.024 / 1.024 /1.024
= 0,05587 GiB/Bild

0,05587 GiB/Bild * 200 Bilder/Woche * 52 Wochen = **581,14 GiB -> 582 GiB**

4. Aufgabe (25 Punkte)

a) 4 Punkte

```
SELECT *
FROM Kunde
WHERE Kd_PLZ LIKE '508%' OR Kd_PLZ LIKE '509%' OR Kd_PLZ LIKE '51%'
```

b) 5 Punkte

```
SELECT Art_Jahrgang AS Jahrgang
        ,MIN(Art_Preis) AS NiedrigsterPreis
        ,MAX(Art_Preis) AS HöchsterPreis
        ,COUNT(Art_IdKey) AS 'Anzahl der Weine je Jahrgang'
FROM Artikel
GROUP BY Art_Jahrgang
ORDER BY Jahrgang DESC;
```

c) 8 Punkte

```
Update      RechnungPosition Set RgPos_RabattProzent = 12
FROM RechnungPosition AS RP
            INNER JOIN Rechnung AS R
                ON R.Rg_IdKey = RP.RgPos_RgIdKey
            INNER JOIN Artikel AS A
                ON A.Art_IdKey = RP.RgPos_ArtIdKey
            INNER JOIN WeinArt AS WA
                ON WA.WA_IdKey = A.Art_WAIdKey
WHERE (Month(R.Rg_Datum) = 5 AND YEAR(R.Rg_Datum) = 2023) AND
            RP.RgPos_RabattProzent = 0 AND
            WA.WA_Weinart = 'Rotwein';
```

d) 8 Punkte

```
SELECT Kd_IdKey, Kd_Firma, Kd_Strasse, Kd_PLZ, Kd_Ort, Kd_Nummer,
        (SELECT AVG(RgPos_Einzelpreis * RgPos_Mg)
            FROM RechnungPosition AS RP
                INNER JOIN Rechnung AS R ON
                    R.Rg_IdKey = RP.RgPos_RgIdKey
            WHERE Kd_IdKey = R.Rg_KdIdKey) AS Umsatz
FROM Kunde;
```